

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims

1. (Original) A system for delivering an electronic document, comprising:

a transcoder proxy coupled to receive the electronic document in a first digital format, wherein the electronic document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event, and wherein the transcoder proxy is configured to:

assign a unique identifier to the element;

form a model of a logical structure of the electronic document;

use the model to produce an original script that includes: (i) at least a portion of the electronic document expressed in a second digital format, and (ii) the element and the identifier assigned to the element; and

a client machine coupled to receive the original script.

2. (Original) The system as recited in claim 1, wherein the client machine is configured to:

use the original script to present the portion of the electronic document;

associate the JAVASCRIPT event with the element;

generate the JAVASCRIPT event in response to user input; and

provide JAVASCRIPT event information and the identifier assigned to the element associated with the JAVASCRIPT event to the transcoder proxy.

3. (Original) The system as recited in claim 1, wherein the model also defines methods for accessing and manipulating the document.
4. (Original) The system as recited in claim 3, wherein the model is a document object model (DOM).
5. (Original) The system as recited in claim 1, wherein the first digital format is a text-based markup language.
6. (Original) The system as recited in claim 5, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).
7. (Original) The system as recited in claim 1, wherein elements of the electronic document are associated with corresponding identifiers within the model, and wherein in response to the JAVASCRIPT event and the identifier provided by the client machine, the transcoder proxy is configured to:

access the element within the model using the identifier;

execute the JAVASCRIPT code, thereby producing a result;

use the model and the result to produce a modification script, wherein the modification script differs from the original script; and

provide the modification script to the client machine.

8. (Original) The system as recited in claim 7, wherein the client machine is coupled to receive the modification script and configured to use the modification script to modify the presented portion of the electronic document.

9. (Original) A transcoder proxy, comprising:

- a synchronous document object model (DOM) generator adapted to receive an electronic document in a first digital format, wherein the electronic document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event, and wherein the synchronous DOM generator comprises an identifier (ID) generator configured to assign a unique identifier to the element, and wherein the synchronous DOM generator is configured to:
  - form a pre-transcoded DOM representing a logical structure of the electronic document, wherein the element is associated with the corresponding identifier within the pre-transcoded DOM;
  - provide a first portion of the electronic document in the first digital format;
- a transcoder coupled to receive the portion of the electronic document in the first digital format and configured to:
  - translate the first portion of the electronic document from the first digital format to an original script in a second digital format, wherein the original script includes the element and the identifier assigned to the element;
  - provide the original script; and
- a JAVASCRIPT engine coupled to the synchronous DOM generator and adapted to receive input JAVASCRIPT event information and an input identifier, wherein the JAVASCRIPT engine is configured to:
  - access an element within the pre-transcoded DOM using the identifier;
  - execute JAVASCRIPT code of the element, thereby producing a result; and
  - provide the result to the synchronous DOM generator.

10. (Original) The transcoder proxy as recited in claim 9, wherein the synchronous DOM generator is further configured to:

use the pre-transcoded DOM and the result to produce a second portion of the electronic document; and

provide the second portion of the electronic document.

11. (Original) The transcoder proxy as recited in claim 10, wherein the transcoder is further configured to:

receive the second portion of the electronic document;

translate the second portion of the electronic document from the first digital format to a modification script in the second digital format; and

provide the modification script.

12. (Original) The transcoder proxy as recited in claim 9, wherein the first digital format is a text-based markup language.

13. (Original) The transcoder proxy as recited in claim 12, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).

14. (Original) A client machine, comprising:

an output device; and

a user agent coupled to the output device and adapted for coupling to a transcoder proxy, wherein the user agent is configured to:

receive an original script from the transcoder proxy, wherein the original script includes an element and an identifier assigned to the element; and

form a transcoded DOM in response to the original script, wherein the transcoded DOM is a representation of the portion of the electronic document.

15. (Currently Amended) The client machine as recited in claim 14, wherein the user agent is further configured to:

use the transcoded DOM to produce output commands;

provide the output commands to the output device;

associate the ~~the~~ JAVASCRIPT event with the element;

generate the JAVASCRIPT event in response to user input;

provide JAVASCRIPT event information and the identifier assigned to the element associated with the JAVASCRIPT event to the transcoder proxy;

receive a modification script from the transcoder proxy;

modify the transcoded DOM in response to the modification script.

16. (Original) The client machine as recited in claim 14, wherein the output device is a display device or a text-to-speech converter.

17. (Original) A system for delivering an electronic document, comprising:

a transcoder proxy, including:

a synchronous document object model (DOM) generator coupled to receive the electronic document in a first digital format, wherein the electronic document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event, and wherein the synchronous DOM generator comprises an identifier (ID) generator configured to assign a unique identifier to each element, and wherein the synchronous DOM generator is configured to:

form a pre-transcoded DOM representing a logical structure of the electronic document, wherein the element is associated with the corresponding identifier within the pre-transcoded DOM; and

provide a portion of the electronic document in the first digital format;

a transcoder coupled to receive the portion of the electronic document in the first digital format and configured to:

translate the portion of the electronic document from the first digital format to an original script in a second digital format, wherein the original script includes the element and the identifier assigned to the element; and

provide the original script;

(  
a JAVASCRIPT engine coupled to the synchronous DOM generator and to receive an input JAVASCRIPT event and an input identifier, wherein the user interface generator is configured to:

access an element within the pre-transcoded DOM using the input identifier;

execute JAVASCRIPT code of the element, thereby producing a result; and

provide the result to the synchronous DOM generator; and

a client machine coupled to receive the original script, wherein the client machine comprises:

an output device;

a user agent coupled to the output device, the transcoder, and the user interface generator, wherein the user agent is configured to:

form a transcoded DOM in response to the original script from the transcoder,  
wherein the transcoded DOM is a representation of the portion of the  
electronic document;

use the transcoded DOM to produce output commands;

provide the output commands to the output device;

associate the JAVASCRIPT event with the element;

generate the JAVASCRIPT event in response to user input; and

provide JAVASCRIPT event information and the identifier assigned to the  
element associated with the JAVASCRIPT event to the JAVASCRIPT  
engine.

18. (Original) The system as recited in claim 17, wherein the synchronous DOM generator is further configured to:

use the pre-transcoded DOM and the result to produce a second portion of the electronic  
document; and

provide the second portion of the electronic document.

19. (Original) The system as recited in claim 18, wherein the transcoder is further configured to:

receive the second portion of the electronic document;

translate the second portion of the electronic document from the first digital format to a  
modification script in the second digital format; and

provide the modification script.

20. (Original) The system as recited in claim 19, wherein the client machine is further configured to:

receive the modification script; and

modify the transcoded DOM in response to the modification script.

21. (Original) The system as recited in claim 17, wherein the first digital format is a text-based markup language.

22. (Original) The system as recited in claim 18, wherein the text-based markup language is hypertext markup language (HTML) or extensible markup language (XML).

23. (Original) A method for transcoding an electronic document having at least one element, comprising:

receiving the electronic document in a first digital format, wherein the document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event;

assigning a unique identifier to the element;

forming a model of a logical structure of the electronic document, wherein the model includes the element and the identifier assigned to the element;

using the model to produce a script, wherein the script includes at least a portion of the document expressed in a second digital format, and wherein the script includes the element and the identifier assigned to the element; and

providing the script.

24. (Original) The method as recited in claim 23, wherein the model also defines methods for accessing and manipulating the document.

25. (Original) The method as recited in claim 24, wherein the model is a document object model (DOM).



26. (Original) The method as recited in claim 23, wherein the first digital format is a text-based markup language.

27. (Original) The method as recited in claim 23, wherein the second digital format is a scripting language.

28. (Original) A method for presenting an electronic document, comprising:

receiving the electronic document in a first digital format, wherein the document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event;

assigning a unique identifier to the element;

forming a model of a logical structure of the electronic document, wherein the model includes the element and the identifier assigned to the element;

using the model to produce an original script, wherein the original script includes at least a portion of the electronic document expressed in a second digital format, and wherein the original script includes the element and the identifier assigned to the element;

providing the original script;

receiving input JAVASCRIPT event information and an input identifier assigned to an element associated with the JAVASCRIPT event;

accessing an element within the model using the input identifier;

executing JAVASCRIPT code of the element, thereby producing a result;

use the model and the result to produce a modification script, wherein the modification script differs from the original script; and

providing the modification script.

29. (Original) A method for presenting an electronic document, comprising:

receiving an original script, wherein the original script includes at least a portion of the electronic document expressed in a digital format, and wherein the original script includes an element of the document, a JAVASCRIPT event associated with the element, and an identifier assigned to the element;

using the original script to present the portion of the electronic document;

associating the JAVASCRIPT event with the element;

generating the JAVASCRIPT event in response to user input;

providing JAVASCRIPT event information and the identifier assigned to the element associated with the JAVASCRIPT event;

receiving a modification script; and

using the modification script to modify the presented portion of the electronic document.

30. (Original) A method for presenting an electronic document, comprising:

receiving the electronic document in a first digital format, wherein the document includes an element, and wherein a JAVASCRIPT event is associated with the element, and wherein the element includes JAVASCRIPT code executed in response to the JAVASCRIPT event;

assigning a unique identifier to the element;

forming a model of a logical structure of the electronic document, wherein the model includes the element and the identifier assigned to the element;

using the model to produce an original script, wherein the original script includes at least a portion of the electronic document expressed in a second digital format, and wherein the original script includes the element and the identifier assigned to the element;

using the original script to present the portion of the document;

associating the JAVASCRIPT event with the element;

generating the JAVASCRIPT event in response to user input;

accessing the element within the model using the identifier;

executing the JAVASCRIPT code of the element, thereby producing a result;

using the model and the result to produce modification script; and

using the modification script to modify the presented portion of the document.